In: 11th European Conference for AeroSpace Sciences (EUCASS 2025), 30 Jun - 4 Jul 2025,
Rome, Italy
https://eucass2025.eu/

Point Cloud Completion based Pose Estimation for Spacecraft

Siddharth Singh[⋆†], *Hyo Sang Shin*[⋆⋆], *Leonard Felicetti*[⋆], *Antonios Tsourdos*[⋆]
[⋆]*Faculty of Engineering and Applied Sciences*
Cranfield University, College Rd, Wharley End, Bedford MK43 0AL, United Kingdom
[⋆⋆] *Cho Chun Shik Graduate School of Mobility, KAIST, Daejeon 34141, Republic of Korea*
siddharth.singh.026@cranfield.ac.uk · hyosangshin@kaist.ac.kr
[†]Corresponding author

Abstract

The growing density of near-Earth space operations necessitates advanced autonomous Guidance, Navigation, and Control (GNC) systems capable of reliable and precise pose estimation during close-proximity operations (CPO). Existing 3D vision-based localization methods, including traditional algorithms and recent machine learning enhancements, often struggle with noisy, sparse, or incomplete point cloud data, limiting their applicability in safety-critical, non-cooperative scenarios. This paper introduces PCC-KIPE, a novel pose estimation framework that overcomes these limitations by combining 3D point cloud completion with a reward-based search algorithm, eliminating the need for exact point correspondences or covariance matrices required by conventional methods. By reconstructing complete target models from partial observations and performing pose estimation through a robust search strategy, PCC-KIPE demonstrates enhanced resilience to data imperfections and reduced reliance on extensive training datasets. The proposed approach offers a promising direction for scalable, autonomous GNC in increasingly complex space environments.

1. Introduction

The increasing density of missions in near-Earth orbits necessitates the development of novel mission architectures that emphasize the sustainable use of space. A substantial number of these missions involve autonomous spacecraft performing close-proximity operations (CPO), such as rendezvous, docking, and in-orbit assembly. As these operations become more routine, they impose stringent requirements on Guidance, Navigation, and Control (GNC) systems, particularly regarding reliability, precision, and operational speed.

A prominent trend in modern GNC design is the integration of autonomous capabilities. This shift mandates highly robust and dependable system architectures capable of operating with minimal human intervention. In this context, we present a non-cooperative pose estimation algorithm that addresses the specific challenges associated with 3D vision-based localization under non-ideal conditions, such as incomplete or noisy point cloud data.

Traditional 3D data analysis methods—such as those by Opromolla[9] and Guohua[5]—perform effectively in well-constrained environments but often require manual tuning, rendering them less suitable for generalized, scalable deployment. Machine Learning (ML) techniques attempt to improve upon these limitations, particularly for dealing with the noise and sparsity typical of LiDAR-based systems. Recent enhancements to the Iterative Closest Point (ICP) algorithm—including Deep Closest Point (DCP),[15] Point Transformer Layer (PTR),[18] and Position-Aware Network (PAN)[6]—leverage ML to generate more accurate point correspondences and covariance matrices, which are then used to estimate pose via the Kabsch-SVD method.[4] However, such methods remain limited by their "black-box" nature and insufficient reliability for safety-critical space applications.

A common shortcoming across these methods is their dependence on accurate point correspondences or covariance estimates. In real-world applications, particularly those involving imperfect or partial point clouds, these assumptions often fail. Since the Kabsch algorithm requires either exact correspondences or a well-defined covariance matrix, even small deviations between source and target data can result in significant pose estimation errors. Furthermore, these methods generally require large, orientation-diverse datasets for training, which may not always be practical. For example, PTR[18] operates within a constrained orientation range, and its performance degrades outside of that domain.

To address the aforementioned limitations, we propose **PCC-KIPE** (Point Cloud Completion + Kabsch Independent

Copyright © 2025 by S. Singh, H.S. Shin, L. Felicetti and A. Tsourdos. Published by the EUCASS association with permission.
Published by EUCASS. This is the Author Accepted Manuscript issued with: Creative Commons Attribution License (CC:BY 4.0).
Please refer to any applicable publisher terms of use.

Pose Estimation) a novel method for non-cooperative pose estimation that bypasses the reliance on deep learning-based correspondence generation or covariance matrix computation. Our approach reconstructs a complete point cloud of the target object from partial observations using a 3D point cloud completion network.[16] This completed structure is then processed by a reward-based search algorithm[1] to determine the most likely pose. Unlike Kabsch-dependent methods, PCC-KIPE does not require precise correspondences or prior covariance estimation, enabling it to maintain robustness under noisy or incomplete data conditions.

Main Contributions

1. We introduce **PCC-KIPE**, a new pose estimation framework tailored for incomplete point clouds. The algorithm integrates 3D completion and reward-based optimization, offering a flight-viable solution.
2. We present a modular and scalable pose estimation pipeline that generalizes to any object with a corresponding CAD model.

2. Related Works

Pose estimation has long been studied in computer vision, traditionally through geometric inference methods. These approaches typically rely on establishing 3D–2D correspondences to regress the object's 6D pose. Classical techniques such as RANSAC with Perspective-n-Point (PnP) utilize keypoint extraction and feature matching to solve for the pose. While conceptually straightforward, such methods are limited by low convergence rates, reduced accuracy with complex geometries, and sensitivity to initialization—factors that make them less suitable for real-time applications.

The ICP algorithm,[14] based on the Kabsch-Umeyama SVD,[4] solves for rotation and translation by iteratively minimizing Euclidean distance and RMSE between point cloud pairs. Variants like PCA-ICP[9] use PCA to estimate an initial pose (roll and pitch from the major axis, yaw from 1D template matching) before refining with ICP. Ying et al.[3] improve upon this by integrating additional geometric features such as curvature and normals, coupled with a RANSAC-ICP fusion for robust estimation. Guohua et al.[5] formulate pose estimation as a nonlinear optimization using a covariance matrix transform and Levenberg-Marquardt,[12] later refined using pose graph optimization.[7]

With the advent of machine learning, newer methods have emerged. Renaut et al.[10] use a CNN (MobileNet backbone) to directly regress pose from 3D data, a "direct" approach relying on soft classification and constrained orientation sampling. DCP[15] introduces a deep geometric module (DGCNN) to generate correspondences for SVD-based pose estimation. Extensions like PTR[18] and PAN[6] build upon this by incorporating transformer-based architectures or saliency-based covariance weighting.

Nonetheless, methods based on the Kabsch-SVD pipeline—including DCP, DCA, PTR, and PAN—perform well only under idealized conditions with complete and clean data. Their accuracy degrades sharply when confronted with real-world noise or incomplete observations. High orientation errors in such conditions have been consistently observed in experimental evaluations (see Fig. 5).

3. Proposed Methodology

We propose a robust two-stage system for 6D pose estimation that synergizes deep learning-based point cloud completion with a numerical pose regression approach. This architecture is specifically designed to handle real-world challenges such as occlusions, incomplete geometry, noisy depth data, and harsh lighting, which are commonly encountered in 3D vision tasks. The pipeline is visualized in Fig.1, and consists of: (1) point cloud completion using a modified Ada-PoinTr network and (2) pose estimation using a reward-based search strategy through PLuM.
We leverage AdaPoinTr,[16] a state-of-the-art transformer-based architecture, for the point cloud completion stage. Given an incomplete point cloud from a 3D sensor (e.g., LiDAR), we first perform Farthest Point Sampling (FPS) to reduce redundancy. This is followed by positional encoding and geometric feature extraction using a DGCNN backbone. The encoded features are passed through a geometry-aware transformer encoder-decoder architecture, enhanced by a dynamic query generator. This network outputs a high-fidelity reconstructed 3D point cloud, even from sparse inputs, benefiting from AdaPoinTrâs adaptability to noise and structural occlusions. Research suggests that completion is achievable even when only 5 to 10% of the object's surface is visible,[11,16,17] and our experiments corroborate this. The completed point cloud is forwarded to the PLuM module,[1] which performs numerical optimization to estimate the 6D pose. This method is Kabsch-independent, thus avoiding dependency on explicit point correspondences. PLuM computes the objectâs centroid (X, Y, Z translation) and its orientation using quaternions (Qx, Qy, Qz). It does so by aligning the completed point cloud with the known object model through a reward-based search strategy, improving robustness under occlusion and view variation.

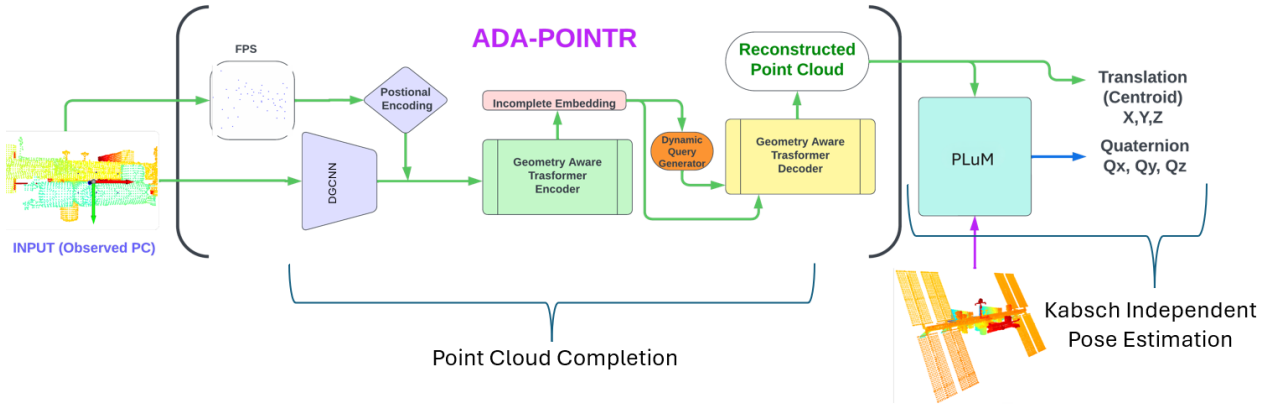


Figure 1: PCC-KIPE : Point Cloud Completion (Ada-PoinTr*) + Search Based PE (PLuM)
Point Cloud Completion + Kabsch Independent Pose Estimation

3.1 Point Cloud Completion : Ada-PoinTr*

After implementing Ada-PoinTr,[16] we encountered discrepancies between the expected inference results and those reported in the original paper. One key challenge was that Ada-PoinTr performs point cloud reconstruction on a normalized scale, where all point coordinates are constrained within the range [-1, 1]. However, this critical detail is not explicitly discussed in the PoinTr paper,[16] particularly in the context of their evaluation results. Upon further investigation into the dataset preprocessing and normalization pipeline used by the algorithm, we identified inconsistencies in the normalization and centralization strategies an issue that appears common across many state-of-the-art (SOTA) methods in point cloud processing. In the following sections, we detail these discrepancies and the modifications we introduced to resolve them.

1. **Centralizing** : For the training of the model, the ground truth data is centred using centroid values from complete point clouds, where as the input data uses the incomplete, therefore incorrect, centroid. In practice, we can only have the incoming depth information, so we overwrite the data normalization step to take the Input centroid only rather than both. We show this change highlighted in bold from Eqs. 1 to 2 below:

$$\begin{aligned} GT_{norm} &= \frac{GT_{points} - Centroid_{GT}}{Scale_{GT}} \\ IP_{norm} &= \frac{IP_{points} - Centroid_{IP}}{Scale_{IP}} \end{aligned} \tag{1}$$

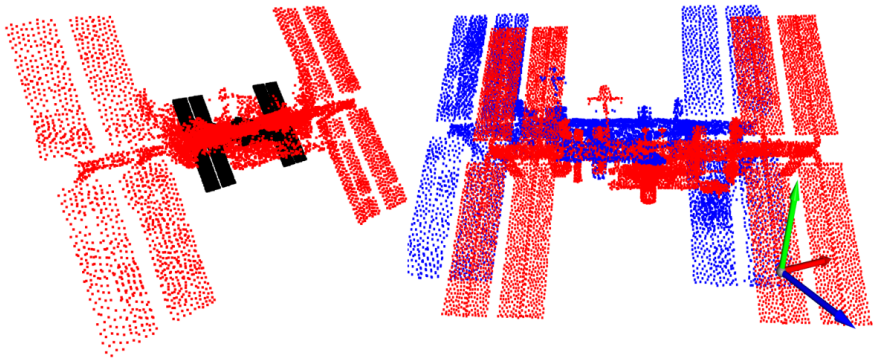
$$\begin{aligned} GT_{norm} &= \frac{GT_{points} - \mathbf{Centroid_{IP}}}{\mathbf{Scale_{Const.}}} \\ IP_{norm} &= \frac{IP_{points} - \mathbf{Centroid_{IP}}}{\mathbf{Scale_{Const.}}} \end{aligned} \tag{2}$$

   where $Centroid_{GT}$ and $Centroid_{IP}$ are the centroids calculated from the ground truth point cloud and the partial input point cloud, GT_{points} and IP_{points} are the point clouds defined in (X, Y, Z), and GT_{norm} and IP_{norm} are the normalized point clouds after post-processing with $Scale_{\text{IP}}$ and $Scale_{\text{GT}}$ being the calculated scaling factors for the input and ground truth respectively.

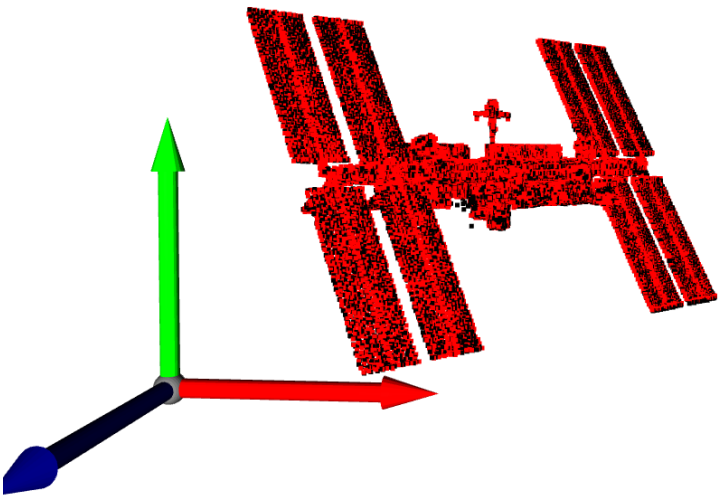
2. **Normalization**: For the base AdaPoinTr algorithm the normalizing factor is a variable calculated at every instant from incomplete input based on the longest distance between points in the input. This value changes with different auto-occlusion scenarios. We can see the errors in reconstruction in Fig.2. The image on the left (*Reconstructed : Red, Ground Through: Black*) shows an error in the scale of reconstruction and loss of fidelity as our learning manifold is not consistent. Taking the ISS model as an example, due to the current normalization strategy the normalizing factor for the model ranges from 3 to 210. Since we are working with SE(3) and rigid body mechanics, we know the scaling factor for a set 3D structure remains constant. The normalizing factor

Scale$_{\text{Const.}}$ in Eq.2 gets the value of 68.75 for the one-to-one ISS model case. Setting the normalization factor as a constant improves overall reconstruction fidelity as we can see in Fig.2.

After implementing the said changes, we call it Ada-PoinTr*, we see improvement in performance in terms of scale, translation and fidelity of point cloud reconstruction for multiple targets, these are displayed in Fig.2. In terms of results, our reconstruction metrics outshine the numbers presented in the paper (discussed in Sec.4). With our reconstructed complete point cloud, we can further use this to regress pose using several different methodologies. To benchmark our proposed method against state of the art we compare our results with current methods like ICP, DCP,[15] PTR.[18] With this comparison we also find reasons for lack of real world applicability with these other algorithms compared to *ours*.



((a)) Errors in Reconstruction using Ada-PoinTr



((b)) Reconstruction using Ada-PoinTr*

Figure 2: Ada-PoinTr v/s Ada-PoinTr*(*ours*)

3.2 Pose Estimation : PLuM

PLuM[1] overcomes the challenges faced by current SOTA by converting the pose estimation problem from an SVD[4] to a reward-based search algorithm, Fig.3. The key idea is to orient the target point cloud back to the GT state from a complete Euler search space, with the highest reward being available at the correct pose.

The reward metric is defined using the Euclidean distance between the points of the target point cloud and the surface of the GT 3D model at the reference orientation. This reward file, lookup table in Fig.3, is generated offline and used to calculate the reward/orientation for the points of the transformed target point cloud. Fig.3 shows the schematic for the Plum Algorithm with visualised lookup tables, and achieved pose estimates. We improve PLuM for our implementation by adding a pose certificate checkpoint based on the reward generation scheme, shown by Eq.3,

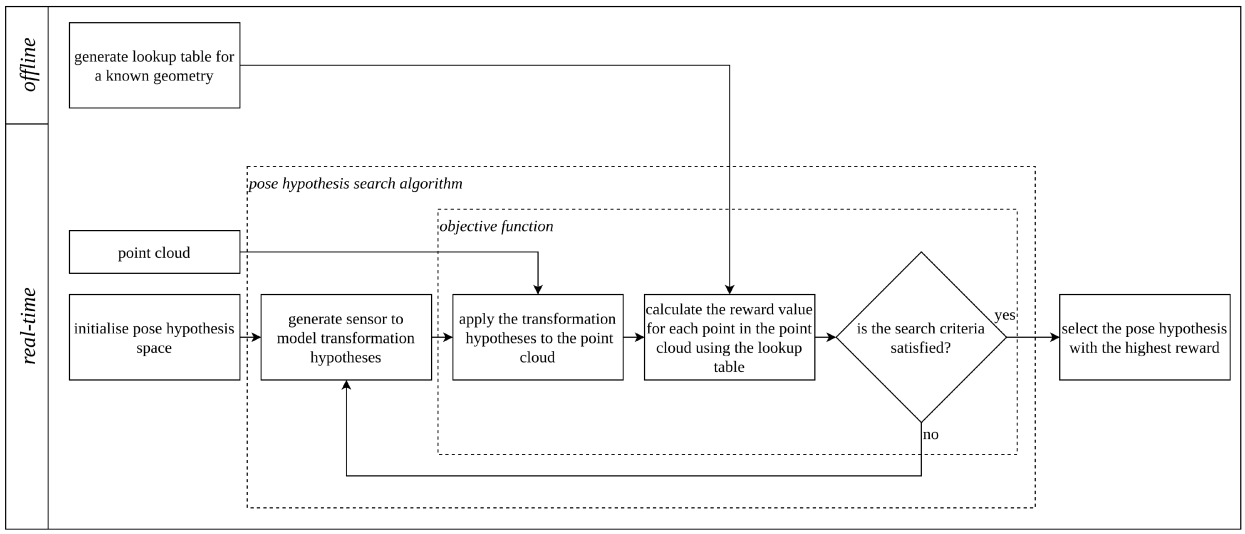


Figure 3: PLuM Algorithm[1]

we set checkpoints for pose as a reward-per-point check. If estimated pose is close to the GT pose, our overall reward value is close to the perfect value. By considering that the maximum reward-per-point is 255, we set the checkpoint value based on the analysis of our experiment and datasets under different configurations, as discussed in Sec. 4.

$$R_{i|j|k} = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(\frac{-d^2_{i|j|k}}{2\sigma^2}\right) \qquad H_i = \sum R_{i|j|k} \tag{3}$$

where $R_{i|j|k}$ is the reward for point [i,j,k], H_i is the reward for all points in the transformation hypothesis I, $d_{i|j|k}$ is the unsigned distance between point [i,j,k] from the GT model surface and σ is the step size of the lookup table.

3.3 Evaluation Metrics

To evaluate our pose estimation system we use Mean Absolute Error in Geodesic Rotation[1] and Euclidean Distance as shown in Eq.4, for both angular and translational errors, respectively. Translation metrics presented are in *meters (m)* and rotational error in *degrees (°)*. We present the obtained mean values along with their standard deviation in Table 3.

$$E_R = 2\arccos(|q_p, q_g|) \qquad E_t = \|T_p - T_g\|_2 \tag{4}$$

where q_p, q_g are the predicted and the ground truth orientations in quaternions and T_p, T_g are the translations in meters, respectively. Going forward in Sec.4 we discuss the results of the proposed changes from Ada-PoinTr to Ada-PoinTr* + PLuM.

3.4 Datasets

To validate our methodology on different shape and sizes of spacecrafts we use publicly available 3D models from NASA Online Resources.[8] We make datasets of 8000+ point cloud samples using Python and Blender[2] for the Ada-PoinTr**(ours)* backbone. We create an additional 1200 samples for testing the complete pipeline per object, i.e., JWST, Dream Chaser, LRO. All training and evaluation is done on NVIDIA RTX A5000 GPU.

[1] https://rotations.berkeley.edu/geodesics-of-the-rotation-group-so3/

4. Experiments and Simulations

For our experiments, we first compare Ada-PoinTr* with the base model for reconstruction results. We discuss these improvements in Sec.4.1. After that, we attach this reconstruction algorithm to ICP, PTR and PLuM to compare pose estimation performance in Sec.4.2.

4.1 Ada-PoinTr*

For all 3D models we generate similar datasets in terms of size and domain of orientations and translations. All training for both Ada-PoinTr and *Ada-PoinTr** is done to specifications mentioned below in Table 1.

Parameter	Value
Epochs	600
Batch Size	16
Learning Rate	0.0002
Optimizer	AdamW
Total Reconstructed Points	13300
Compute	10.5 GFLOPS[16]

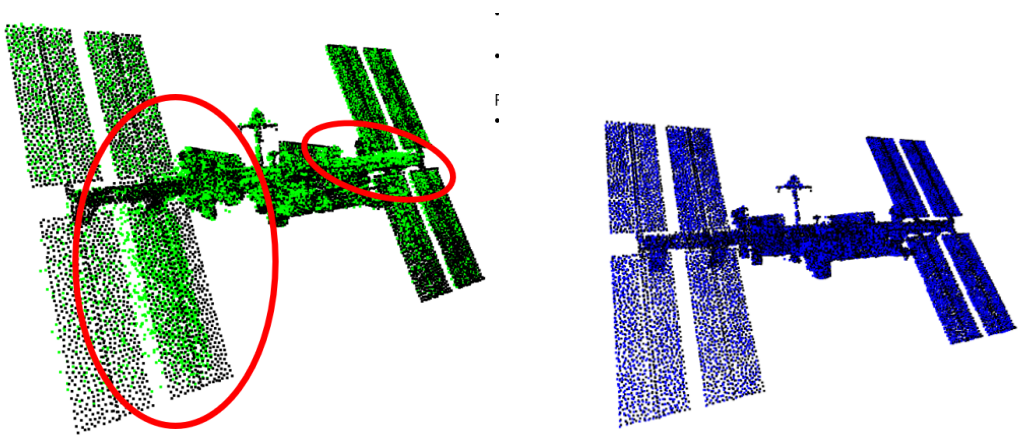
Table 1: Training parameters for Point Cloud reconstruction

To show improvement in reconstruction fidelity we show the metrics Chamfer Distances (CDL1, CDL2) and F-Score in Table 2, over a test-set of 1000+ point clouds per spacecraft type. For the ISS dataset, which is the most structurally complex and large in scale compared to rest of the point clouds, we can see that there is almost a 30% increase in F-score performance from the base model to ours, with even greater increase in reconstruction fidelity for both CDL1 and CDL2 metrics.

Model	Chamfer Distance L1 (x1000)	Chamfer Distance L2 (x1000)	F-Score
Ada-PoinTr (base)	8.3082	0.458	0.740
Ada-PoinTr * *(Ours)*	5.7 **(31.3%)**	0.083 **(82.8%)**	0.960 **(29.7%)**

Table 2: Comparing Reconstruction Fidelity between Ada-PoinTr v/s Ada-PoinTr*

Figure 4 also highlights the visual change in fidelity of reconstruction with Ada-PoinTr in green on the left and Ada-PoinTr* in blue on the right, compared to the ground truth in black. The highlighted areas show an increase in fidelity of reconstruction in similar places, with a clear boost in performance of reconstruction for Ada-PoinTr*. We highlight this increase in performance because of the perfect-imperfect correspondence problem faced by DCP, PTR and ICP (ref. Sec3.2), and we see how this affects the performance of these algorithms.



((a)) Ada-PoinTr ((b)) Ada-PoinTr*

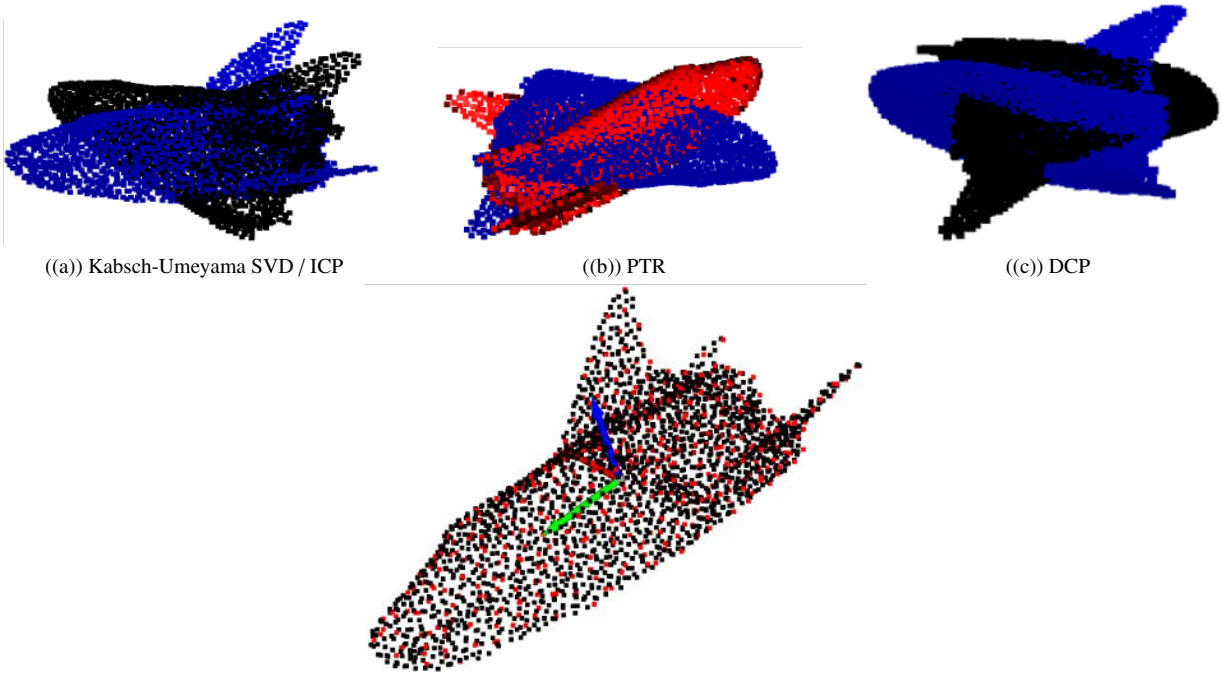
Figure 4: Fidelity Comparison : Ada-PoinTr v/s Ada-PoinTr*(*ours*)

4.2 SOTA v/s Ada-PoinTr* + PLuM

To benchmark our solution, we use the same two metrics for 6D Pose Estimation, as in Eq.4. It is worth noting that the DCP and PTR were not able to match the ground truth with sufficient accuracy, as shown in Fig.5. For comparison, we use the results published in their paper for James Web Space Telescope (JWST), Lunar Reconnaissance Orbiter (LRO), EUCLID and Dream Chaser/Space Shuttle. With our initial tests using Ada-PoinTr* + ICP our errors were between 10°to 30°, which highlights the hypotheses of errors existing with the implementation of the Kabsch-SVD.[4] Current SOTA and through NASA provided literature,[13] we see that the current International Docking Adapter (IDA) allows a maximum of 3°vector sum over pitch and yaw misalignment during the final docking phase. We use this value to set as a minimum benchmark for our system. Keeping these results in mind and problems with the SVD solution discussed in Sec.3.2, in Table 3 we compare registration results of JWST and LRO from PTR[18] as a benchmark and share the results for Dream Chaser and EUCLID as the argument for robust performance with different types and sizes of spacecraft. All experiments use point clouds of 2048 points.

Spacecraft	Angular Error°	Translation Error (m)	ADD-S (5%)
JWST	$0.965 \pm 0.545 \mid 1.180_{PTR}$	$0.23 \mid 0.019_{PTR}$	1
LRO	$1.334 \pm 0.688 \mid 0.916_{PTR}$	$0.06 \mid 0.006_{PTR}$	1
Dream Chaser	0.811 ± 0.125	0.069	1
EUCLID	0.927	-	1

Table 3: Results of Pose Estimation using *our* pipeline v/s PTR[18]



((a)) Kabsch-Umeyama SVD / ICP ((b)) PTR ((c)) DCP

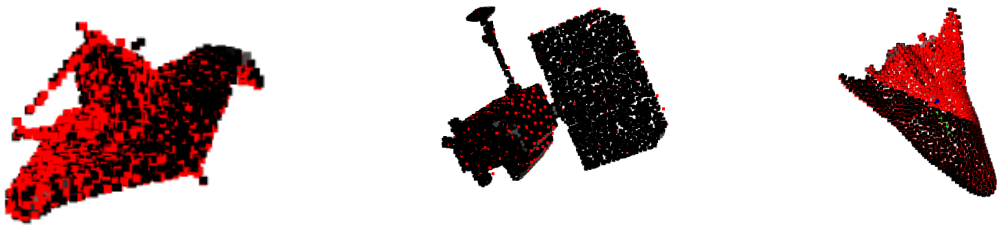
((d)) PLuM

Figure 5: Pose Estimation on Dream Chaser with Imperfect Correspondences (Ada-PoinTr* Output) *Target : Blue, Pose Estimate: Black/Red*

Fig.6 shows examples of different spacecraft and their poses estimated using Ada-PoinTr* + PLUM. With our improvements to PLuM, we cover the entire domain of Intrinsic XYZ rotation with high fidelity in the first search pass, while reducing size and adding noise to the search space.

4.3 Ablation Study

To understand and validate the improvement made over the original PLUM algorithm we run the same test with our implemented *pose certificate* and the base algorithm over the Dream Chaser dataset. The improvements made with the



((a)) JWST ((b)) LRO ((c)) Dream Chaser

Figure 6: Examples of Pose Estimated *Target : Black, Pose Estimate: Red*

base Ada-PoinTr are crucial to reach this step as shown in Fig.4, since the PLUM algorithm depends on the fidelity of overall shape reconstructed. In Table 4 we can see that our pose accuracy and mean variance improves significantly over the base algorithm. The results presented in Table 3 are the results of extended improvements over the complete PLUM algorithm, which will be discussed later in an expanded journal submission.

Certificate	Dream Chaser
×	8.831 ± 15.243
✓	1.500 ± 7.505

Table 4: Ablation study of Modifications made to PLuM for Dream Chaser

5. Conclusions

In this work, we presented a novel two-stage pipeline for robust 6D pose estimation that integrates transformer-based point cloud completion with a reward-based numerical pose solver.

In contrast to these hybrid and learning-heavy approaches, such as CMT, PCA-ICP, PAN, PTR/DCP etc., our method takes a fundamentally different direction. We adopted PLuM,[1] a robust alternative that replaces the SVD decomposition with a reward-based numerical search, eliminating the dependence on point correspondences and neural network predictions for alignment. To ensure that this pose estimation operates on complete 3D information, we also addressed the challenge of partial observations by modifying and optimizing Ada-PoinTr. This approach enabled high-fidelity point cloud reconstruction in real-time. Unlike existing methods that assume complete input data, our pipeline actively reconstructed the full geometry before pose estimation, improving resilience to occlusions and noisy sensor inputs.

With these key innovations replacing fragile ML-based alignment with a search-based method and completing partial inputs through a robust transformer backbone, we believe we have developed a novel, scalable, and reliable 6D pose estimation pipeline. This framework is specifically suited for spacecraft pose estimation across a wide range of structural complexities and scales, offering both generalization and accuracy without manual tuning or post-refinement steps.

5.1 Future Work

In future expansion of this work, we plan to publish all the changes implemented to the Base Ada-PoinTr and PLUM Algorithm. We also do a thorough ablation study with overall compute times, compute memory and accuracy over different spacecrafts. We validate the methodology for other previously studied natural satellites (asteroids or comets). We also release all data generation and post processing libraries to enable quick custom data-generation setup for other future researches and researchers.

6. Acknowledgments

This research was supported by the Inha University funded project Precision Guidance and Navigation for Deep Space Exploration (project number P20962).

References

[1] Vedant Bhandari, Tyson Govan Phillips, and Peter Ross McAree. Real-Time 6-DOF Pose Estimation of Known Geometries in Point Cloud Data. *Sensors*, 23(6), 2023.

[2] Blender Foundation. Blender 3.5, 2023. 3D Creation Software.

[3] Ying He, Jun Yang, Kejiang Xiao, Chao Sun, and Jia Chen. Pose tracking of spacecraft based on point cloud dca features. *IEEE Sensors Journal*, 22(6):5834–5843, 2022.

[4] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, Sep 1976.

[5] Guohua Kang, Qi Zhang, Jiaqi Wu, and Han Zhang. Pose estimation of a non-cooperative spacecraft without the detection and recognition of point cloud features. *Acta Astronautica*, 179:569–580, 2021.

[6] Xiang Liu, Hongyuan Wang, Xinlong Chen, Weichun Chen, and Zhengyou Xie. Position awareness network for noncooperative spacecraft pose estimation based on point cloud. *IEEE Transactions on Aerospace and Electronic Systems*, 59(1):507–518, 2023.

[7] Ellon Mendes, Pierrick Koch, and Simon Lacroix. Icp-based pose-graph slam. In *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 195–200, 2016.

[8] NASA. Nasa 3d resources, 2025.

[9] Roberto Opromolla, Giancarmine Fasano, Giancarlo Rufino, and Michele Grassi. Pose estimation for spacecraft relative navigation using model-based algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 53(1):431–447, 2017.

[10] Leo Renaut, Heike Frei, and Andreas Nuchter. Cnn-based pose estimation of a non-cooperative spacecraft with symmetries from lidar point clouds. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–16, 2024.

[11] Rahul Sajnani, Adrien Poulenard, Jivitesh Jain, Radhika Dua, Leonidas J. Guibas, and Srinath Sridhar. Condor: Self-supervised canonicalization of 3d pose for partial shapes, 2022.

[12] Dewi Mutiara Sari and Vina Wahyuni Eka Putranti. Optimization estimating 3d object pose using levenberg-marquardt method. In *2019 International Electronics Symposium (IES)*, pages 582–589, 2019.

[13] Scott P. Cryan Sean M. Kelly. International docking standasrt (idss) interface definition document (did) : Revison-e. Technical report, NASA Johnson SPace Center Houston, TX, United States, 2017.

[14] Fang Wang and Zijian Zhao. A survey of iterative closest point algorithm. In *2017 Chinese Automation Congress (CAC)*, pages 4395–4399, 2017.

[15] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration, 2019.

[16] Xumin Yu, Yongming Rao, Ziyi Wang, Jiwen Lu, and Jie Zhou. Adapointr: Diverse point cloud completion with adaptive geometry-aware transformers, 2023.

[17] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling, 2022.

[18] Shaodong Zhang, Weiduo Hu, and Wulong Guo. 6-dof pose estimation of uncooperative space object using deep learning with point cloud. In *2022 IEEE Aerospace Conference (AERO)*, pages 1–7, 2022.

https://dspace.org/

Point cloud completion based pose estimation for spacecraft

Singh, Siddharth; Shin, Hyo-Sang; Felicetti, Leonard; Tsourdos, Antonios

2025-07-04
EUCASS
https://doi.org/10.13009/EUCASS2025-615
Singh S, Shin H-S, Felicetti L, Tsourdos A. (2025) Point cloud completion based pose estimation for spacecraft. In: 11th European Conference for AeroSpace Sciences (EUCASS 2025), 30 Jun - 4 Jul 2025, Rome, Italy

Downloaded from a DSpace repository. https://dspace.org/